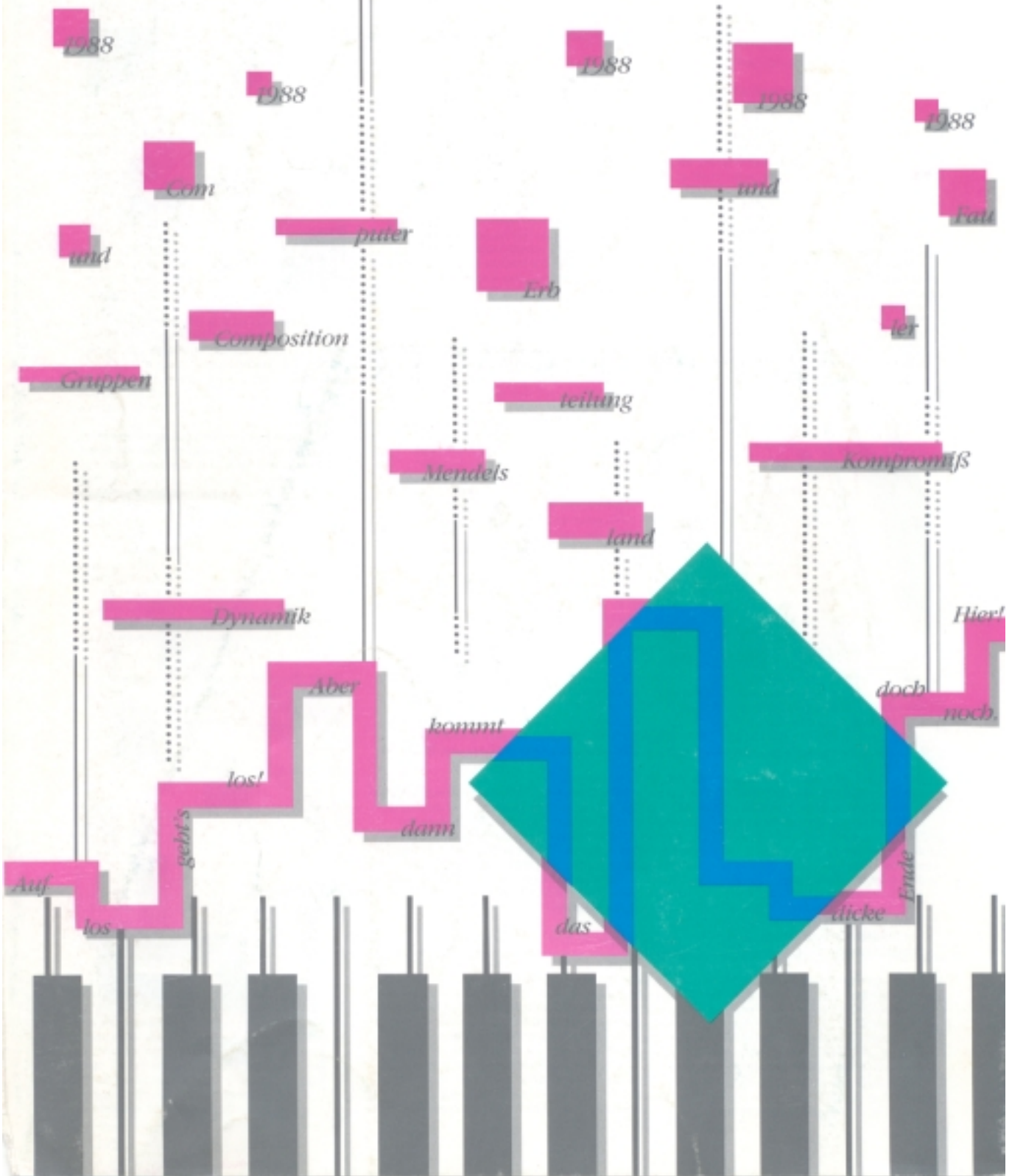


7. DAS AUFGABENBLATT

BUNDES WETTBEWERB INFORMATIK



Aufgabe 1: Computer-Composition

Sue Dama ist berühmt für ihre anspruchsvollen Kompositionen. Sie benutzt immer nur die fünf Tonstufen C, D, F, G und A. Zunächst gibt sie sich ein Thema aus n gleichlangen Tönen vor. Jede Tonstufe kann dabei mehrfach, einfach oder gar nicht vorkommen. Ein Thema für $n=4$ ist zum Beispiel: „AGCA“.

Weiter denkt sie sich eine Abbildungsregel zwischen den Tonstufen aus, deren Anwendung auf das Thema automatisch eine Variation ergibt. Die Abbildungsregel

„C→F, D→D, F→D, G→C, A→G“

ergibt für das BeispieltHEMA die Variation: „GCFG“.

Jetzt wählt sie noch eine Satzform, die aus einer beliebigen Aneinanderreihung von Thema und Variation besteht. Für die Satzform „Thema-Variation-Variation-Thema“ sieht der erste Satz ihrer Komposition im Beispiel nun so aus: „AGCA GCFG GCFG AGCA“.

Den zweiten Satz komponiert Sue Dama, indem sie den ganzen ersten Satz als Thema nimmt, und die Abbildungsregel und die Satzform darauf wiederholt. Für das Beispiel lautet der zweite Satz damit:

„AGCAGCFGGGCFGAGCA
GCFGCFDCCFCGCFG
GCFGCFDCCFCGCFG
AGCAGCFGGGCFGAGCA“.



Weil der zweite Satz viel mehr Töne hat als der erste, spielt ihn Sue Dama um einen Prestofaktor schneller. Ist der Prestofaktor etwa gleich 2,0, dann dauern alle Töne nur halb so lang wie im vorhergehenden Satz.

Der dritte, noch schnellere Satz ergibt sich aus dem zweiten, ebenso wie der zweite aus dem ersten Satz. Der vierte Satz ergibt sich analog aus dem dritten, usw.

Aufgabe:

Schreibe ein Programm, das im Stil von Sue Dama komponiert, wenn man ihm ein Thema, eine Abbildungsregel, eine Satzform, die Anzahl der gewünschten Sätze, die Tondauer im ersten Satz und einen Prestofaktor eingibt.

Schicke uns mindestens 5 Kompositionen mit mehreren Sätzen. Eines der Beispiele sei das folgende: Thema „AGCA“, Abbildungsregel „C→F, D→D, F→D, G→C, A→G“, Satzform „Thema-Variation-Variation-Thema“, 4 Sätze, Tondauer im ersten Satz 4 Zeiteinheiten, Prestofaktor 1,7.

Aufgabe 2: Fauler Kompromiß

In der Politik fertigt man nach zähen Verhandlungen, die nicht zu einer Übereinstimmung der beiden Parteien geführt haben, oft ein Schlußkommuniqué an, in dem immerhin möglichst viele gemeinsame Positionen dargelegt werden. Um nicht auch noch über das Schlußkommuniqué lange verhandeln zu müssen, sollte man verabreden, daß zunächst jede Partei einen Textvorschlag macht. Als Schlußkommuniqué wird dann der längste Text gewählt, der als Teil beider Textvorschläge auftritt. Dabei werden der Einfachheit halber nur ganze Wörter und Satzzeichen betrachtet, nicht aber einzelne Buchstaben.

Beispiel: Aus den Textvorschlägen

„Die Einkommen der Landwirte sind Abgeordneten ein Buch mit sieben Siegeln; um dem abzuhelfen, müssen dringend alle Subventionsgesetze verbessert werden.“

und

„Die Steuern auf Vermögen und Einkommen sollten nach Meinung der Abgeordneten nachdrücklicher erhoben werden. Dazu müssen die Kontrollbefugnisse der Finanzbehörden dringend verbessert werden.“

würde das Schlußkommuniqué erstellt:

„Die Einkommen der Abgeordneten müssen dringend verbessert werden.“

Aufgabe:

Schreibe ein Programm, das nach Eingabe von zwei Textvorschlägen, deren Länge nicht festgelegt ist, ein solches Schlußkommuniqué ermittelt.

Schicke uns mindestens 5 Paare von Textvorschlägen und die dazugehörigen Schlußkommunikés. Eines der Textpaare sei:

„Viele landwirtschaftliche Familienbetriebe kämpfen um ihre Existenz. Der deutsche Bauernverband hat den Bundeskanzler sehr bestimmt aufgefordert, die Europäische Gemeinschaft auf höchster Ebene zu veranlassen, die diesbezüglichen Richtlinien den Notwendigkeiten der landwirtschaftlichen Politik der Bundesregierung anzugleichen.“

und

„Der Agrarausschuß der EG berücksichtigt bisher deutsche Belange nach Auffassung aller landwirtschaftlicher Organisationen nur unzureichend. Daher sollen Bundeskanzler und Landwirtschaftsminister behutsam aber bestimmt die Abänderung einiger Richtlinien für Agrarpreise im Sinne der Politik der Bundesregierung bewirken. Viele landwirtschaftliche Betriebe leben von ihrer Substanz. Einige Familienbetriebe kämpfen bereits verzweifelt um ihre Existenz.“

Aufgabe 3: Erbteilung

Die Baronin von Birlinghoven hat ihren beiden Töchtern eine Truhe voller Goldmünzen hinterlassen. Ihr Testament bestimmt, daß das Gold einem benachbarten Kloster,



zukommt, falls es den Töchtern nicht gelingt, den Inhalt der Truhe wertmäßig genau in zwei Hälften untereinander aufzuteilen. Die Goldmünzen haben nur ganzzahlige Werte.

Beispiel:

Eine Truhe Goldmünzen mit den Werten 1, 9, 5, 3, 8 Taler könnten die Töchter in die Hälften 1, 9, 3 Taler und 5, 8 Taler teilen.

Eine Truhe Goldmünzen mit den Werten 1, 9, 7, 3, 8 Taler fiel an das Kloster, weil die Aufteilung nicht möglich ist.

Aufgabe:

Schreibe ein Programm, das bei Eingabe einer Folge ganzer Zahlen für die in der Truhe vorkommenden Werte die beiden Erben genau aufzählt, andernfalls das Erbe dem Kloster zuspricht, wenn eine Aufteilung nicht möglich ist.

Schicke uns mindestens 5 Beispiele mit verschiedenen Truheneinhalten. Der Inhalt einer Truhe sei: 15, 27, 39, 7, 23, 56, 13, 39, 22, 5, 42, 34 Taler.



Aufgabe 4: Gruppendynamik

Stehparties erfreuen sich nicht zuletzt deswegen steigender Beliebtheit, weil man mit seinem Glas in der Hand im Raum umherwandern kann, bis man ein angenehmes Plätzchen gefunden hat. Wolfi beispielsweise versucht stets, bis auf drei Schritte an Sabine heranzukommen; sich noch weiter anzunähern wäre für ihn gesellschaftlich nicht akzeptabel. Sabine dagegen bleibt möglichst weit von Wolfi entfernt, allerdings nicht weiter als 15 Schritte, damit er sie noch sehen kann. So gibt es für jeden Gast eine Wunschentfernung zu jedem anderen Gast.

Ein Partygast ist wunschlos glücklich, wenn er sich zu allen anderen Gästen in seiner Wunschentfernung befindet. Je stärker die tatsächlichen Entfernungen von der Wunschentfernung abweichen, desto weniger glücklich ist er. Das kollektive Partyglück ergibt sich als Durchschnitt des Glücks aller Gäste.

Solange noch Gäste glücklicher werden können, ist Bewegung in der Party. Dann wandert nämlich jeder in die Richtung, die sein Glück am meisten steigert. Nimm an, daß der rechteckige Raum in 20x40 quadratische Felder eingeteilt ist, und daß jeder Partygast mit einem Schritt von dem Feld, in dem er sich gerade befindet, in eines der acht benachbarten Felder gelangen kann. Es können natürlich nur freie Felder betreten werden. Nimm weiter über den Verlauf der Party an, daß in einer Runde reihum jeder der Gäste einen Schritt machen oder auch stehenbleiben darf.



Nach jeder Runde wird das kollektive Glück berechnet, wobei alle Entfernungen in Schritten ausgedrückt werden. Die Party ist zu Ende, wenn alle Gäste stehengeblieben sind.

Aufgabe:

Schreibe ein Programm, das nach Eingabe der Anzahl der Partygäste, der zugehörigen Wunschentfernungen in Schritten sowie einer Anfangsaufstellung der Gäste die Bewegung in der Party am Bildschirm zeigt. Es soll protokollieren, wie sich das kollektive Glück im Verlauf der Party entwickelt. Schicke uns die Protokolle von mindestens fünf verschiedenen Parties.

Aufgabe 5: Mendels Land

In Mendels Land gibt es eine fantastische Vielfalt von Schmetterlingen. Man sieht welche mit roten schwarz gepunkteten Flügeln und gekrümmten Fühlern, andere sind schwarzgelb gestreift und haben gerade Fühler usw.



Bei längerer Beobachtung können wir drei Typen von Merkmalen unterscheiden:

1. Musterung: uni, schwarz gepunktet oder schwarz gestreift.
2. Flügelfarbe: rot, gelb, grün oder blau.
3. Fühlerform: gerade oder gekrümmt.

Es stellt sich heraus, daß jeder Schmetterling pro Merkmalstyp ein dominantes Merkmal (das sieht man) und ein weiteres rezessives Merkmal (das sieht man nicht oder es ist gleich dem ersten) in sich trägt. Es gelten folgende Dominanzregeln:

| | | |
|-------------------|-----------|------------------------------------------|
| uni | dominiert | schwarz gepunktet und schwarz gestreift, |
| schwarz gepunktet | dominiert | schwarz gestreift, |
| rot | dominiert | grün und blau, |
| gelb | dominiert | rot und blau, |
| grün | dominiert | gelb und blau, |
| gerade | dominiert | gekrümmt. |

Ein Schmetterlingsei erbt für jeden Merkmalstyp von beiden Eltern zufällig je eines von deren zwei Merkmalen. Die in dieser neuen Kombination dominanten Merkmale bestimmen dann das Aussehen des späteren neuen Schmetterlings. Zum Beispiel:

| | |
|----------------------------|-------------------------------------|
| Mutter: sichtbar: | uni – rot – gerade |
| nicht sichtbar: | schwarz gestreift – blau – gekrümmt |
| Vater: sichtbar: | uni – grün – gerade |
| nicht sichtbar: | schwarz gepunktet – grün – gekrümmt |
| Kind: zufällig von Mutter: | schwarz gestreift – blau – gekrümmt |
| zufällig von Vater: | uni – grün – gekrümmt |
| sichtbar: | uni – grün – gekrümmt |

Aufgabe:

Schreibe ein Programm, das bei Eingabe der dominanten Merkmale zweier Eltern und der gewünschten Kinderzahl entsprechend viele Kinder „mendelt“ und beschreibt. Die rezessiven Merkmale der Eltern werden vom Programm zufällig, aber unter Beachtung der Dominanzregeln hinzugefügt.

Schicke uns mindestens 5 Beispiele von verschiedenen Elternpaaren mit ihrer Nachkommenschaft.

Allgemeine Hinweise

Alle Einsendungen sollten aus den folgenden Teilen bestehen, **getrennt nach Aufgaben:**

Lösungsidee:

Eine Beschreibung der Lösungsidee. Die Form und die Begriffe der Lösungsidee müssen sich im Programm wiederfinden.

Programm-Dokumentation:

Eine Beschreibung des Programms. Hinweise auf Besonderheiten und Nutzungsgrenzen. Angaben zu Rechenzeiten und Speicherbedarf.

Programmablauf-Protokoll:

Kommentierte Probeläufe des Programms. Manchmal genügt ein Protokoll, manchmal müssen es mehrere sein.

Programm-Text:

Das Programm selbst in einer der gängigen Programmiersprachen. Keinen Assembler benutzen. Der Text sollte eine Zeilennummerierung haben.

Deine Einsendungen werden danach bewertet

- ob sie vollständig und richtig sind,
- ob die Ausarbeitungen gut strukturiert und verständlich sind,
- ob die Programmunterlagen übersichtlich und lesbar sind.

Bitte schicke Deine Ergebnisse in einem Exemplar, auf einseitig beschriebenen oder bedruckten DIN-A4-Blättern. Alle Blätter sind rechts oben durchzunummerieren und mit Deinem vollständigen Namen zu versehen. Die Einsendungen müssen in deutscher Sprache abgefaßt sein. Sende uns keine Disketten ein.

Bitte fülle das Begleitformular (die abtrennbare Klappe am Aufgabenblatt oder eine Kopie davon) vollständig aus. Bei Gruppen muß jeder Teilnehmer ein Begleitformular ausfüllen und unterschreiben. Die statistischen Angaben haben keinen Einfluß auf die Bewertung.

Stecke das Exemplar Deiner Ergebnisse zusammen mit dem Begleitformular in einen DIN-A4-Umschlag mit der Adresse:

Bundswettbewerb Informatik 1988
GMD-Schloß Birlinghoven
5205 Sankt Augustin 1

Einsendeschluß ist der 28. November 1988 (Datum des Poststempels gilt). Bei mehreren Einsendungen wird nur die vor Einsendeschluß zuletzt abgeschickte bewertet. Verspätete Einsendungen werden nicht geöffnet. Der Rechtsweg ist ausgeschlossen.

Die Einsendungen werden nicht an die Teilnehmer zurückgegeben. Mit der Einsendung wird den Veranstaltern des Wettbewerbs das Recht übertragen, die Beiträge in geeigneter Form zu veröffentlichen.

Teilnehmen können Jugendliche, die nach dem 28. November 1966 geboren wurden. Sie dürfen jedoch im Sommersemester 1988 noch nicht studiert oder vor dem 1. September 1988 ihre Ausbildung beendet und einen Beruf ergriffen haben (Näheres siehe unter „Teilnahmeberechtigung“).

**Musteraufgabe:
Goldgräberclaim**

Ein Goldgräber hat nach Streifzügen durch das weite, unerforschte Land einige vielversprechende Schürflplätze entdeckt. Diese Plätze hat er auf einer geheimgehaltenen Karte vermerkt.

Bevor er jedoch mit seinen Grabungen beginnen kann, muß er das von ihm beanspruchte Gebiet, seinen Claim, nach alter Goldgräbersitte abstecken. Dazu will er einen Zaun errichten, der den Claim um-

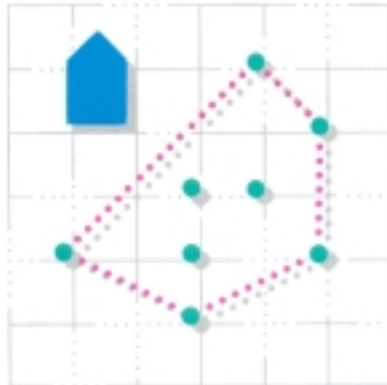


Abbildung A

schließt. Weil das Material teuer ist, soll der Zaun möglichst kurz sein. Zum Beispiel ist der kürzeste Zaun um den in der Abbildung A gezeigten Claim aus acht Schürflplätzen 12,13 Rastereinheiten lang.

Aufgabe:

Schreibe ein Programm, das nach Eingabe beliebig vieler Schürflplätze die Streckenbeschreibung und die Länge des kürzesten Claim-Zaunes ermittelt und ausgibt. Jeder Schürflplatz wird durch ein Paar (x,y) ganzzahliger Koordinaten angegeben. Zur Vereinfachung kannst Du davon ausgehen, daß ein Schürflplatz keine Ausdehnung besitzt und auch dann schon zum Claim gehört, wenn er genau unter dem Zaun zu liegen kommt.

Schicke uns mindestens 5 Zaunberechnungen. Einer der Zäune sei für einen Claim mit folgenden Schürflplätzen: (8,17) (9,16) (12,18) (9,19) (10,17) (7,16) (9,15).

**Musterlösung: Nr. 4
vom Aufgabenblatt 1987**

Lösungsidee

Die abstrakte Formulierung des Problems ist: „Finde zu einer gegebenen Menge von Punkten in einer Ebene die konvexe Hülle in Gestalt eines Polygons. Gib die zyklische Folge der Punkte aus, die Ecken dieses Polygons sind.“

Zunächst wird ein Startpunkt gesucht, der offensichtlich zur Hülle gehört (Linksunterster_Punkt) und dann das Polygon links herum (gegen den Uhrzeiger, in mathematisch positiver Drehrichtung) berechnet. Dazu wird jeweils das Minimum der Winkel zu allen anderen Punkten gesucht (Nächstaeusserster_Punkt). Gibt es mehrere Punkte mit minimalem Winkel, wird der am weitesten entfernte genommen; dieser ist Ecke des Polygons und neuer Startpunkt. Der Vorgang wird abgebrochen, wenn der linksunterste Punkt wieder erreicht ist.

Das Minimum der Winkel kann bestimmt werden, ohne diese selbst zu berechnen: Für einen Punkt A = (x1, y1) und zwei Kandidaten B = (x2, y2) und C = (x3, y3) hat Punkt B genau dann den kleineren Winkel, wenn die Fläche des Dreiecks ABC negativ ist. Die Formel ergibt sich, wenn man z. B. in Abb. B die drei schraffierten rechtwinkligen Dreiecke vom umschreibenden Rechteck abzieht. Sie gilt für beliebige Dreiecke.

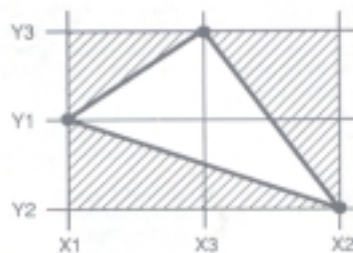


Abbildung B

Die PROCEDURE Ausgabe_der_Zaunbeschreibung folgt genau der Lösungsidee. Sie benutzt die FUNCTION Linksaeusserster_Punkt zur Berechnung des ersten Polygonpunktes (Zeile 53) und FUNCTION Nächstaeusserster_Punkt zur Berechnung des jeweils links herum nächsten Polygonpunktes (Zeile 57).

Programm-Dokumentation

Die drei globalen Konstanten können bei Bedarf vergrößert werden, ohne daß sonst etwas am Programm verändert werden muß.

Die PROCEDURE Eingabe_der_Schuerflplaetze wickelt die interaktive Dateneingabe ab (aus Platzgründen wurde die direkte Prüfung der Daten, ob sie in den zulässigen Wertebereichen liegen, weggelassen).

Die FUNCTION Linksunterster_Punkt berechnet die Nummer des Punktes mit der kleinsten X-Koordinate (Zeile 28). Liegen mehrere Punkte gleichweit links, dann wird der mit der kleinsten Y-Koordinate genommen.

Die FUNCTIONs Dreiecksflaeche und Streckenlaenge entsprechen den bekannten geometrischen Formeln.

Programmablauf-Protokoll

Zaunberechnung Nr. 1:
Koordinaten der Schürflplätze eingeben. Jeweils eine Zahl für x, ein Leerzeichen (blank), eine Zahl für y, dann das Endezeichen (return).
Werte für x zwischen 1 und 40 und für y zwischen 1 und 20. Beenden durch Eingabe der Koordinaten "0 0".

- 1: 8 17
- 2: 9 16
- 3: 12 18
- 4: 9 19
- 5: 10 17
- 6: 7 16
- 7: 9 15
- 8: 0 0

Koordinaten der Eckpfosten des Claimzaunes:

- 1: 7 16
- 2: 9 15
- 3: 12 18
- 4: 9 19

Länge des Claimzaunes: 13.25

Es folgen Zaunberechnung 2 bis mindestens 5 mit jeweils anderen Daten (hier aus Platzgründen weggelassen).

Programm-Text

Das beliegende Programm ist in TurboPascal formuliert und auf Microcomputern unabhängig von einem bestimmten Betriebssystem ablauffähig.

```

1 PROGRAM Goldgraeberclaim; { 8. Bundeswettbewerb Informatik 1987/1988 }
2 { 1. Runde, Aufgabe 4, Musterlösung Pascal }
3
4 CONST maximale_x_Koordinate = 40; {maximale = 1}
5 CONST maximale_y_Koordinate = 20; {maximale = 1}
6 CONST maximale_Anzahl_Schuerflplaetze = 100; {maximale = 1}
7 VAR X, Y : ARRAY [1..maximale_Anzahl_Schuerflplaetze] OF INTEGER;
8     Anzahl_Punkte : INTEGER;
9
10 PROCEDURE Eingabe_der_Schuerflplaetze; VAR Zaehler : INTEGER;
11 BEGIN write(' Koordinaten der Schürflplätze eingeben. Jeweils eine Zahl ');
12 write(' für x, ein Leerzeichen (blank), eine Zahl für y, dann die ');
13 write(' Endezeichen (return). ');
14 write(' Werte für x zwischen 1 und ', maximale_x_Koordinate, ');
15 write(' und für y zwischen 1 und ', maximale_y_Koordinate, ' ');
16 write(' Beenden durch Eingabe der Koordinaten "0 0" ');
17 Zaehler := 0; write(' ');
18 REPEAT Zaehler := Zaehler + 1; write(Zaehler, ' ');
19 readln(X[Zaehler], Y[Zaehler]);
20 UNTIL (X[Zaehler] = 0) and (Y[Zaehler] = 0);
21 Anzahl_Punkte := Zaehler - 1;
22 END;
23
24 FUNCTION Linksaeusserster_Punkt : INTEGER;
25 VAR Markierung, Zaehler : INTEGER;
26 BEGIN Markierung := 1;
27 FOR Zaehler := 2 to Anzahl_Punkte DO
28 IF ( X[Zaehler] < X[Markierung] ) OR
29 ( X[Zaehler] = X[Markierung] AND
30 ( Y[Zaehler] < Y[Markierung] ) )
31 THEN Markierung := Zaehler;
32 Linksaeusserster_Punkt := Markierung; END;
33
34 FUNCTION Dreiecksflaeche ( A, B, C : INTEGER) : REAL;
35 BEGIN Dreiecksflaeche := 0.5 * (X[A]*Y[B] - X[B]*Y[C] + X[C]*Y[A]
36 - X[A]*Y[C] - X[B]*Y[A] - X[C]*Y[B]); END;
37
38 FUNCTION Streckenlaenge ( A, B : INTEGER) : REAL;
39 BEGIN Streckenlaenge := sqrt ( sq ( X[A] - X[B] ) + sq ( Y[A] - Y[B] ) ); END;
40
41 FUNCTION Nächstaeusserster_Punkt ( A : INTEGER) : INTEGER;
42 VAR Markierung, Zaehler : INTEGER;
43 BEGIN Markierung := 1;
44 FOR Zaehler := 2 to Anzahl_Punkte DO
45 IF ( Dreiecksflaeche ( A, Markierung, Zaehler ) < 0 ) OR
46 ( Dreiecksflaeche ( A, Markierung, Zaehler ) = 0 ) AND
47 ( Streckenlaenge ( A, Markierung ) < Streckenlaenge ( A, Zaehler ) )
48 THEN Markierung := Zaehler;
49 Nächstaeusserster_Punkt := Markierung; END;
50
51 PROCEDURE Ausgabe_der_Zaunbeschreibung;
52 VAR A, B, Zaehler : INTEGER; Zaehleng : REAL;
53 BEGIN A := Linksaeusserster_Punkt; Zaehler := 0; Zaehleng := 0; write(' ');
54 write(' Koordinaten der Eckpfosten des Claimzaunes: '); write(' ');
55 REPEAT Zaehler := Zaehler + 1;
56 write( Zaehler, ' ', X[A], ' ', Y[A], ' ');
57 B := Nächstaeusserster_Punkt ( A );
58 Zaehleng := Zaehleng + Streckenlaenge ( A, B );
59 A := B;
60 UNTIL A = Linksaeusserster_Punkt;
61 write(' '); write(' Länge des Claimzaunes: ', Zaehleng, ' ');
62 END;
63
64 BEGIN Eingabe_der_Schuerflplaetze; Ausgabe_der_Zaunbeschreibung; END;

```